UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/567,851	10/06/2006	Jouni Kytomaa	39700-638N01US/NC40070US	7339

64046 7590 08/31/2010

MINTZ, LEVIN, COHN, FERRIS, GLOVSKY AND POPEO, P.C
ONE FINANCIAL CENTER
BOSTON, MA 02111

EXAMINER	
MITCHELL, DANIEL D	

ART UNIT	PAPER NUMBER
2477	

MAIL DATE	DELIVERY MODE
08/31/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/567,851	KYTOMAA ET AL.	
	Examiner	Art Unit	
	DANIEL MITCHELL	2477	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 June 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-17 and 35-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-17, 35-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 February 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's amendment filed on 6/26/2010 has been entered. Claims 1, 2, 35, and 46 have been amended. Claims 18-34 are canceled. Claims 1-17 and 35-46 are still pending in this application, with claims 1, 2, 35, and 45 being independent.

Response to Arguments

2. Applicant's arguments with respect to claim1-46 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-8, 11-17, 35-38, and 41-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ma et a. (US Patent No. 6,832,265 B1), hereinafter referred as Ma in view of Gorti et al. (US Patent No. 7,385,997 B2), hereinafter referred as Gorti in view of LaVigne et al (US Publication No. 2003/0169757 A1), hereinafter referred as LaVigne.

Regarding claim 1, Ma teaches wherein each received packet comprises an internet protocol packet **(col. 1 lines 16-33 teaches receiving a layer 3 packet)**;

responsive to transfer of a packet to a transfer queue, generating an interrupt **(col. 2 lines 30-43 teaches generating an interrupt to the processor when a packet is transferred to the transfer queue)**;

responsive to receipt of an interrupt, allocating the packet from said transfer queue to one of a plurality of processor queues **(col. 2 line 30-43 teaches responsive to an interrupt, the packet is allocated to the proper software queue (processor queue).**

placing the packet in the allocated processor queue if said processor queue is not full, **(col. 2 lines 30-43 teaches placing the packet in the software queue)** and Ma teaches a plurality of processor queues (software queues) are separate queues **(col. 2 lines 30-43, fig. 1).**

However Ma does not expressly disclose allocating each received packet to at least one arrival queue; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet; scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one transfer queue; scheduling packets from the processor queues to be processed, wherein the at least one arrival queue, the at least one transfer queue, and the plurality of processor queues are separate queues, wherein the scheduler includes a first quantity N of inputs each corresponding to

the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N.

Gorti teaches in **fig. 2, col. 7 lines 18-49** scheduling, by a scheduler coupled to the at least one arrival queue **[queues 203A-B]**, packets from the arrival queue to at least one transfer queue **[queue 205] (fig. 2, col. 7 lines 18-49 teaches a scheduler 204 that is coupled to the arrival queues 203A-B and transfer queue 205)** and Gorti further teaches dropping packets based on the queue condition (abstract),

scheduling packets from the processor queues to be processed, wherein the at least one arrival queue, and the at least one transfer queue **(fig. 2 col. 7 lines 18-49 teaches scheduling packets to be processed and also teaches a separate arrival queue and separate transfer queue)**,

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N **(fig. 2, col. 7 lines 18-49 teaches the scheduler (204) includes a first quantity of inputs from the arrival queues (203A-B) being greater than the single output of the scheduler to the transfer queue (205))**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a

scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

However Ma and Gorti do not expressly disclose allocating each received packet to at least one arrival queue, placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet.

LaVigne teaches allocating each received packet to at least one arrival queue **(par. 20, 21, fig. 2 teaches allocating the received packet to at least one arrival queue)**; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet **(par. 20, 21 teaches placing the packet in the arrival queue; par. 9 teaches dropping packets)**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma and Gorti to include allocating input buffers. One would be motivated as such in order to apply a priority to the received packet (par. 11).

Regarding claim 2, Ma teaches a method comprising: receiving a plurality of packets at one or more of a plurality of network devices coupled via one or more buses to at least one of a processor and a memory **(col. 8 lines 29-54 teaches a processor and memory of the devices)**, wherein the packet comprises an internet protocol packet **(col. 1 lines 16-33 teaches receiving a layer 3 packet)**;

responsive to transfer of a packet to the at least one transfer queue, generating an interrupt **(col. 2 lines 30-43 teaches generating an interrupt to the processor when a packet is transferred to the transfer queue)**;

responsive to receipt of the interrupt, allocating the packet from the at least one transfer queue to at least one of the plurality of processor queues **(col. 2 line 30-43 teaches responsive to an interrupt, the packet is allocated to the proper software queue (processor queue)**;

placing the packet in the allocated processor queue if said processor queue is not full, **(col. 2 lines 30-43 teaches placing the packet in the software queue)** and Ma teaches a plurality of processor queues (software queues) are separate queues **(col. 2 lines 30-43, fig. 1).**

However Ma does not expressly disclose allocating each received packet to at least one arrival queue, wherein the one or more of the plurality of network devices comprises a scheduler, the at least one arrival queue, and at least one transfer queue, wherein the at least one of the processor and the memory further comprises at least one of a plurality of processor queues; wherein the scheduler is further coupled to the at least one transfer queue; placing each packet in the allocated arrival queue if the at least one arrival queue is not full, otherwise dropping the packet; scheduling, by the scheduler coupled to the at least one arrival queue, each packet from the arrival queue to the at least one transfer queue; scheduling packets from the at least one of the plurality of processor

queues to be processed, wherein the at least one arrival queue, the at least one transfer queue, and the plurality of processor queues are separate queues

Gorti teaches **fig. 2, col. 7 lines 18-49** wherein the one or more of the plurality of network devices comprises a scheduler **[scheduler 204]**, the at least one arrival queue, and at least one transfer queue **(fig. 2 teaches the arrival queue, transfer queue and scheduler)**,

wherein the scheduler is further coupled to the at least one transfer queue **(fig. 2, col. 7 lines 18-49 teaches a scheduler 204 that is coupled to the transfer queue 205)**;

scheduling, by the scheduler coupled to the at least one arrival queue, each packet from the arrival queue to the at least one transfer queue **(fig. 2, col. 7 lines 18-49 teaches a scheduler 204 that is coupled to the arrival queues 203A-B and transfer queue 205 for scheduling received packets)**;

scheduling packets to be processed, wherein the at least one arrival queue and the at least one transfer queue are separate queues **(fig. 2 col. 7 lines 18-49 teaches scheduling packets to be processed and also teaches a separate arrival queue and separate transfer queue).**

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

However Ma and Gorti does not expressly disclose placing each packet in the allocated arrival queue if the at least one arrival queue is not full, otherwise dropping the packet.

LaVigne teaches allocating each received packet to at least one arrival queue **(par. 20, 21, fig. 2 teaches allocating the received packet to at least one arrival queue)**; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet **(par. 20, 21 teaches placing the packet in the arrival queue; par. 9 teaches dropping packets)**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma and Gorti to include allocating input buffers. One would be motivated as such in order to apply a priority to the received packet (par. 11).

Regarding claim 3, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein at least one device has a plurality of arrival queues.

Gorti teaches a plurality of arrival queues in fig. 2, col. 7 lines 18-49 teaches a device with a plurality of arrival queues 203A-B.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a

scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

Regarding claim 4, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein each arrival queue is associated with a traffic class, each packet being allocated to at least one queue by the processor in accordance with the traffic class of each packet.

Gorti teaches a plurality of arrival queues in fig. 2, col. 7 lines 18-49 where each queue is associated with a traffic class (i.e. real-time and non-real-time classes).

See similar motivation as claim 3.

Regarding claim 5, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma does not expressly disclose wherein the traffic class is priority information embedded in the each packet.

Gorti teaches in col. 5 lines 31-52 determining the class of a packet by examining a priority value of the packet.

See similar motivation as claim 3.

Regarding claim 6, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein at least one device comprises a plurality of transfer queues.

Gorti teaches the transfer which can contain a plurality of transfer queues (col. 7 lines 18-49).

See similar motivation as claim 3.

Regarding claim 7, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma does not expressly disclose wherein the number of transfer queues is less than the number of arrival queues.

Gorti teaches in fig. 2 col. 7 lines 18-49 the number of transfer queue [queues 205] is les than the number of arrival queues [queues 203A-B].

See similar motivation as claim 3.

Regarding claim 8, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma does not expressly disclose wherein the scheduling of packets from the arrival queue to the transfer queue is dependent upon one or more of: the quality of service requirement.

Gorti teaches in col. 7 lines 18-49 the scheduling of the packets from the arrival queues to the transfer queue is based on the quality of service requirement of the packet.

See similar motivation as claim 3.

Regarding claim 11, Ma teaches wherein packets are never dropped from the transfer queue **(col. 2 lines 3-6 teaches instead of dropping packets, packets are moved to the processor queue)**.

Regarding claim 12, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein the queues are associated with different priorities.

Gorti teaches in col. 7 lines 18-49 where the queues are associated with different priorities.

See similar motivation as claim 3.

Regarding claim 13, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein the highest priority queue has the lowest drop probability and the lowest latency.

Gorti teaches in col. 7 lines 18-49 the real-time queue which suggests the priority given to real time queue is that having a lowest drop probability and lowest latency.

See similar motivation as claim 3

Regarding claim 14, Ma teaches wherein responsive to receipt of the interrupt, a packet is removed from a transfer queue and classified **(col. 2 lines 30-43 teaches removing packets from the transfer queue responsive to the interrupt and classified)**.

Regarding claim 15, Ma and Gorti and LaVigne teaches a method as the parent claim.

However Ma does not expressly disclose wherein the classification is based on a determination of priority.

Gorti teaches in col. 7 lines 18-49 classification of the packets is based on a priority. Gorti teaches the two priorities are real-time and non-real-time.

See similar motivation as claim 3.

Regarding claim 16, Ma and Gorti and LaVigne teaches a method as the parent claim

However Ma does not expressly disclose wherein 16 wherein the packet is allocated to a processor queue in accordance with a classification of the packet.

Gorti teaches in col. 7 lines 18-49 the queue is allocated in accordance with the classification of the packet. Gorti further teaches the different priority queues.

See similar motivation as claim 3.

Regarding claim 17, Ma teaches wherein the packet is placed in the allocated processor queue if said queue is not full, otherwise the packet is dropped **(col. 2 lines 30-43 teaches placing the packet in the software queue)**.

Regarding claim 35, Ma teaches a processor **col. 8 lines 44-46;** wherein each received packet comprises an internet protocol packet **(col. 1 lines 16-33 teaches receiving a layer 3 packet)**;

responsive to transfer of a packet to a transfer queue, generating an interrupt **(col. 2 lines 30-43 teaches generating an interrupt to the processor when a packet is transferred to the transfer queue)**;

responsive to receipt of an interrupt, allocating the packet from said transfer queue to one of a plurality of processor queues **(col. 2 line 30-43 teaches responsive to an interrupt, the packet is allocated to the proper software queue (processor queue).**

placing the packet in the allocated processor queue if said processor queue is not full, **(col. 2 lines 30-43 teaches placing the packet in the**

software queue) and Ma teaches a plurality of processor queues (software queues) are separate queues **(col. 2 lines 30-43, fig. 1).**

However Ma does not expressly disclose allocating each received packet to at least one arrival queue; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet; scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one transfer queue; scheduling packets from the processor queues to be processed, wherein the at least one arrival queue, the at least one transfer queue, and the plurality of processor queues are separate queues, wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N.

Gorti teaches in **fig. 2, col. 7 lines 18-49** scheduling, by a scheduler coupled to the at least one arrival queue **[queues 203A-B]**, packets from the arrival queue to at least one transfer queue **[queue 205] (fig. 2, col. 7 lines 18-49 teaches a scheduler 204 that is coupled to the arrival queues 203A-B and transfer queue 205)** and Gorti further teaches dropping packets based on the queue condition (abstract),

scheduling packets from the processor queues to be processed, wherein the at least one arrival queue, and the at least one transfer queue **(fig. 2 col. 7**

lines 18-49 teaches scheduling packets to be processed and also teaches a separate arrival queue and separate transfer queue),

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N **(fig. 2, col. 7 lines 18-49 teaches the scheduler (204) includes a first quantity of inputs from the arrival queues (203A-B) being greater than the single output of the scheduler to the transfer queue (205))**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

However Ma and Gorti do not expressly disclose allocating each received packet to at least one arrival queue, placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet.

LaVigne teaches allocating each received packet to at least one arrival queue **(par. 20, 21, fig. 2 teaches allocating the received packet to at least one arrival queue)**; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet **(par. 20, 21 teaches placing the packet in the arrival queue; par. 9 teaches dropping packets)**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma and Gorti to include allocating input buffers. One would be motivated as such in order to apply a priority to the received packet (par. 11).

Regarding claim 36, Ma and Gorti and LaVigne teaches an apparatus as the parent claim.

However Ma does not expressly disclose wherein at least one device has a plurality of arrival queues.

Gorti teaches a plurality of arrival queues in fig. 2, col. 7 lines 18-49 teaches a device with a plurality of arrival queues 203A-B.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

Regarding claim 37, Ma and Gorti and LaVigne teaches an apparatus as the parent claim.

However Ma does not expressly disclose wherein each arrival queue is associated with a traffic class, each packet being allocated to at least one queue by the processor in accordance with the traffic class of each packet.

Gorti teaches a plurality of arrival queues in fig. 2, col. 7 lines 18-49 where each queue is associated with a traffic class (i.e. real-time and non-real-time classes).

See similar motivation as claim 36.

Regarding claim 38, Ma and Gorti and LaVigne teaches an apparatus as the parent claim.

However Ma does not expressly disclose wherein at least one device comprises a plurality of transfer queues.

Gorti teaches the transfer which can contain a plurality of transfer queues (col. 7 lines 18-49).

See similar motivation as claim 36.

Regarding claim 41, Ma teaches wherein packets are never dropped from the transfer queue **(col. 2 lines 3-6 teaches instead of dropping packets, packets are moved to the processor queue)**.

Regarding claim 42, Ma and Gorti and LaVigne teaches an apparatus as the parent claim.

However Ma does not expressly disclose wherein the queues are associated with different priorities.

Gorti teaches in col. 7 lines 18-49 where the queues are associated with different priorities.

See similar motivation as claim 36.

Regarding claim 43, Ma teaches configured responsive to receipt of the interrupt, to remove a packet from a transfer queue, and to classify the packet **(col. 2 lines 30-43 teaches removing packets from the transfer queue responsive to the interrupt and classified)**.

Regarding claim 44, Ma and Gorti and LaVigne teaches an apparatus as the parent claim.

However Ma does not expressly disclose wherein the classification is based on a determination of priority.

Gorti teaches in col. 7 lines 18-49 classification of the packets is based on a priority. Gorti teaches the two priorities are real-time and non-real-time.

See similar motivation as claim 36.

Regarding claim 45, Ma teaches wherein the packet is placed in the allocated processor queue if said queue is not full, and otherwise the packet is dropped **(col. 2 lines 30-43 teaches placing the packet in the software queue)**.

Regarding claim 46, Ma teaches a non-transitory computer-readable storage medium encoded with instructions that, when executed on a computer **(col. 8 lines 54-58 teaches a storage medium)**, perform a process, the process comprising:

wherein each received packet comprises an internet protocol packet **(col. 1 lines 16-33 teaches receiving a layer 3 packet)**;

responsive to transfer of a packet to a transfer queue, generating an interrupt **(col. 2 lines 30-43 teaches generating an interrupt to the processor when a packet is transferred to the transfer queue)**;

responsive to receipt of an interrupt, allocating the packet from said transfer queue to one of a plurality of processor queues **(col. 2 line 30-43 teaches responsive to an interrupt, the packet is allocated to the proper software queue (processor queue).**

placing the packet in the allocated processor queue if said processor queue is not full, **(col. 2 lines 30-43 teaches placing the packet in the software queue)** and Ma teaches a plurality of processor queues (software queues) are separate queues **(col. 2 lines 30-43, fig. 1).**

However Ma does not expressly disclose allocating each received packet to at least one arrival queue; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet; scheduling, by a scheduler coupled to the at least one arrival queue, packets from the arrival queue to at least one transfer queue; scheduling packets from the processor

queues to be processed, wherein the at least one arrival queue, the at least one transfer queue, and the plurality of processor queues are separate queues, wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N.

Gorti teaches in **fig. 2, col. 7 lines 18-49** scheduling, by a scheduler coupled to the at least one arrival queue **[queues 203A-B]**, packets from the arrival queue to at least one transfer queue **[queue 205] (fig. 2, col. 7 lines 18-49 teaches a scheduler 204 that is coupled to the arrival queues 203A-B and transfer queue 205)** and Gorti further teaches dropping packets based on the queue condition (abstract),

scheduling packets from the processor queues to be processed, wherein the at least one arrival queue, and the at least one transfer queue **(fig. 2 col. 7 lines 18-49 teaches scheduling packets to be processed and also teaches a separate arrival queue and separate transfer queue)**,

wherein the scheduler includes a first quantity N of inputs each corresponding to the at least one arrival queue, the scheduler further including a second quantity M of outputs each corresponding to the at least one transfer queue, wherein the second quantity M is less than or equal to the first quantity N **(fig. 2, col. 7 lines 18-49 teaches the scheduler (204) includes a first**

quantity of inputs from the arrival queues (203A-B) being greater than the single output of the scheduler to the transfer queue (205)).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma to include utilizing a scheduler. One would be motivated as such in order to provide priority to data by managing the congestion of data in the network (abstract).

However Ma and Gorti do not expressly disclose allocating each received packet to at least one arrival queue, placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet.

LaVigne teaches allocating each received packet to at least one arrival queue **(par. 20, 21, fig. 2 teaches allocating the received packet to at least one arrival queue)**; placing each packet in the allocated arrival queue if said arrival queue is not full, otherwise dropping said packet **(par. 20, 21 teaches placing the packet in the arrival queue; par. 9 teaches dropping packets)**.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma and Gorti to include allocating input buffers. One would be motivated as such in order to apply a priority to the received packet (par. 11).

5. Claims 9, 10, 39, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ma, Gorti, and LaVigne in view of Vinnakota (US Patent No. 6,789,056).

Regarding claim 9, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma, Gorti and LaVigne do not expressly disclose wherein the transfer queue comprises a device level transfer queue and a processor level transfer queue, wherein the device level transfer queue receives packets from the arrival queue, and the processor level transfer queue receives packets from the device level transfer queue.

Vinnakota teaches in col. 5 lines 3-14 a transfer queue with a device level transfer queue and a processor level transfer queue. The RAM 122 serves as the device level transfer queues and the queues 140-150 serves as the processor level transfer queues. Vinnakota further teaches in col. 5 lines 3-14 the processor level queues receives packets from the device level queue.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma, Gorti and LaVigne to include generating an interrupt signal when a packet is received at a transfer queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

Regarding claim 10, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma, Gorti and LaVigne do not expressly disclose wherein packets are transferred to the processor level transfer queue from the device

level transfer queue whenever there is space in the processor level transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 3-14 that when available packets are transferred from the device level queue to the processor level transfer queue.

See similar motivation as claim 9.

Regarding claim 39, Ma and Gorti and LaVigne teach an apparatus as the parent claim.

However Ma, Gorti and LaVigne do not expressly disclose wherein the transfer queue comprises a device level transfer queue and a processor level transfer queue, wherein the device level transfer queue receives packets from the arrival queue, and the processor level transfer queue receives packets from the device level transfer queue.

Vinnakota teaches in col. 5 lines 3-14 a transfer queue with a device level transfer queue and a processor level transfer queue. The RAM 122 serves as the device level transfer queues and the queues 140-150 serves as the processor level transfer queues. Vinnakota further teaches in col. 5 lines 3-14 the processor level queues receives packets from the device level queue.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Ma, Gorti and LaVigne to include generating an interrupt signal when a packet is received at a transfer

queue. One would be motivated as such in order minimize the processing performed by a packet processor col. 3 lines 11-23.

Regarding claim 40, Ma and Gorti and LaVigne teach a method as the parent claim.

However Ma, Gorti and LaVigne do not expressly disclose wherein packets are transferred to the processor level transfer queue from the device level transfer queue whenever there is space in the processor level transfer queue.

Vinnakota teaches a packet processor as the primary reference (col. 3 lines 11-23). Vinnakota teaches in col. 5 lines 3-14 that when available packets are transferred from the device level queue to the processor level transfer queue.

See similar motivation as claim 39.

Conclusion

6. Any response to this action should be **faxed** to (571) 173-8300 or **mailed** to:

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand delivered responses should be brought to:
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DANIEL MITCHELL whose telephone number is (571)270-5307. The examiner can normally be reached on Monday - Friday 8:00 am - 5:00 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chirag G. Shah can be reached on 571-272-3144. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/D. M./
Examiner, Art Unit 2477

/Chirag G Shah/
Supervisory Patent Examiner, Art Unit 2477